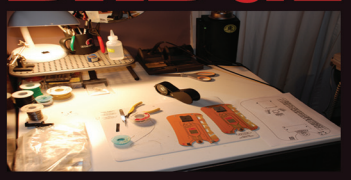
MAKING THE DEFCON 16 BADGE



by Joe Grand

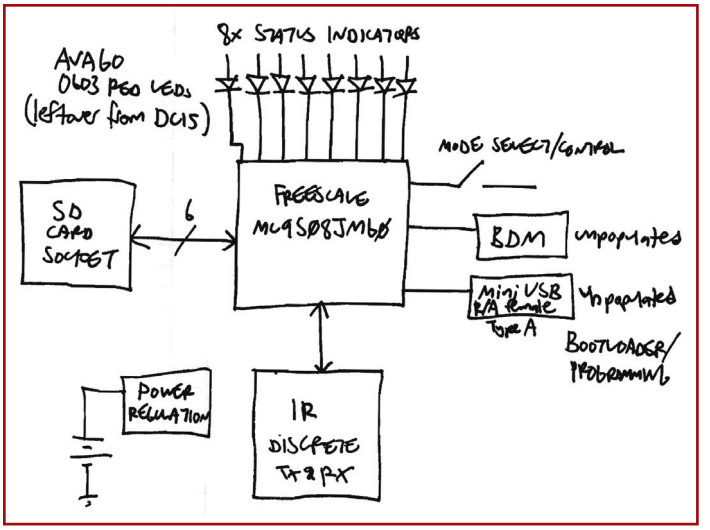
Every summer, thousands of hackers and computer security enthusiasts descend into Las Vegas for DEFCON (www.defcon.org) — the largest and oldest continuously running event of its kind. It's a mix of good guys, bad guys, government officials, and everyone in between, all focused on having fun, sharing technical information, seeing old friends, and learning new things. This is the third year in a row that I've had the honor of designing the conference badge for DEFCON. Unlike other conferences where boring plastic or metal badges are used, DEFCON has been setting the trend since 2006 in giving out full-featured, active electronic badges to their attendees and challenging them to do something unique with their new-found technology.

This article highlights my design process and the problems that I encountered during the creation of the DEFCON 16 Badge. Hopefully, you'll be able to learn from my mistakes or build on my work to enhance your own endeavors.

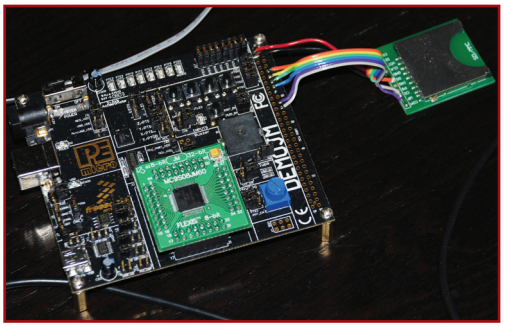
A Brief History of the DEFCON Badge

The previous years' electronic badge designs each had their own set of unique challenges, interesting lessons, and frustrating problems. The DEFCON 14 badge was a round PCB with complicated cutouts of graphical elements and consisted of a six pin Microchip PIC10F202, two jumbo blue LEDs, and a single CR2032 Lithium coin cell. The badge had four different LED modes (on, blinking, alternating, random) and a Microchip ICD2 programming interface for attendees to load their own customized firmware onto the badge. We didn't know what to expect when we started handing them out to conference attendees, but the response was overwhelming, which led to a new badge design for the next year.

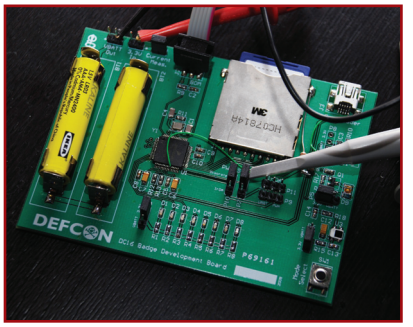
I upped the ante and the technical complexity of the DEFCON 15 badge by using a Freescale MC9S08QG8, a 95 LED matrix (five columns by 19 rows) for custom scrolling text messages, capacitive touch sensors, and



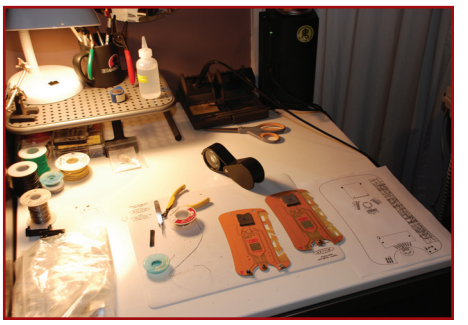
■ FIGURE 1. Preliminary system-level block diagram of the DEFCON 16 badge.



■ FIGURE 2. Freescale's DEMOJM evaluation board with an added SD card socket used for my initial development.



■ FIGURE 3. Custom DEFCON 16 badge development board. The majority of hardware and firmware development was done on this platform before moving to the "form and function" pre-production prototype.



■ FIGURE 4. Using two hand-soldered pre-production prototypes let me verify that the complete design looked and worked as desired. Each board took about one hour to assemble.

unpopulated areas for accelerometer and 802.15.4 wireless support. You can read all about the trials and tribulations of the DEFCON 15 badge in the July 2008 issue of *Nuts & Volts*. Even with the success of the badge, I felt that I had over-engineered it and that it contained too much for attendees to digest over the weekend. So with DEFCON 16, we wanted to have an electronic badge that could still be personalized in some way, but without a lot of "noise" to detract people from the key features and turn them off from hacking their badge.

Design Goals

The primary goal of the DEFCON 16 badge was to incorporate a file transfer feature to allow an attendee to transfer files to another attendee using his or her badge, not unlike the "beaming" capability of PDAs and smartphones. Attendees would load their desired file — be it a business card, picture, poem, or write-up of their latest discovery or research — onto a SecureDigital (SD) card, insert it into the badge, and transfer it to a willing recipient via infrared. Just like last year's badge in which attendees could display a customized text message onto it, the file transfer functionality of DEFCON 16 would meet this same "personalization" goal, allowing attendees to make their badge unique based on what sort of information they chose to share with others.

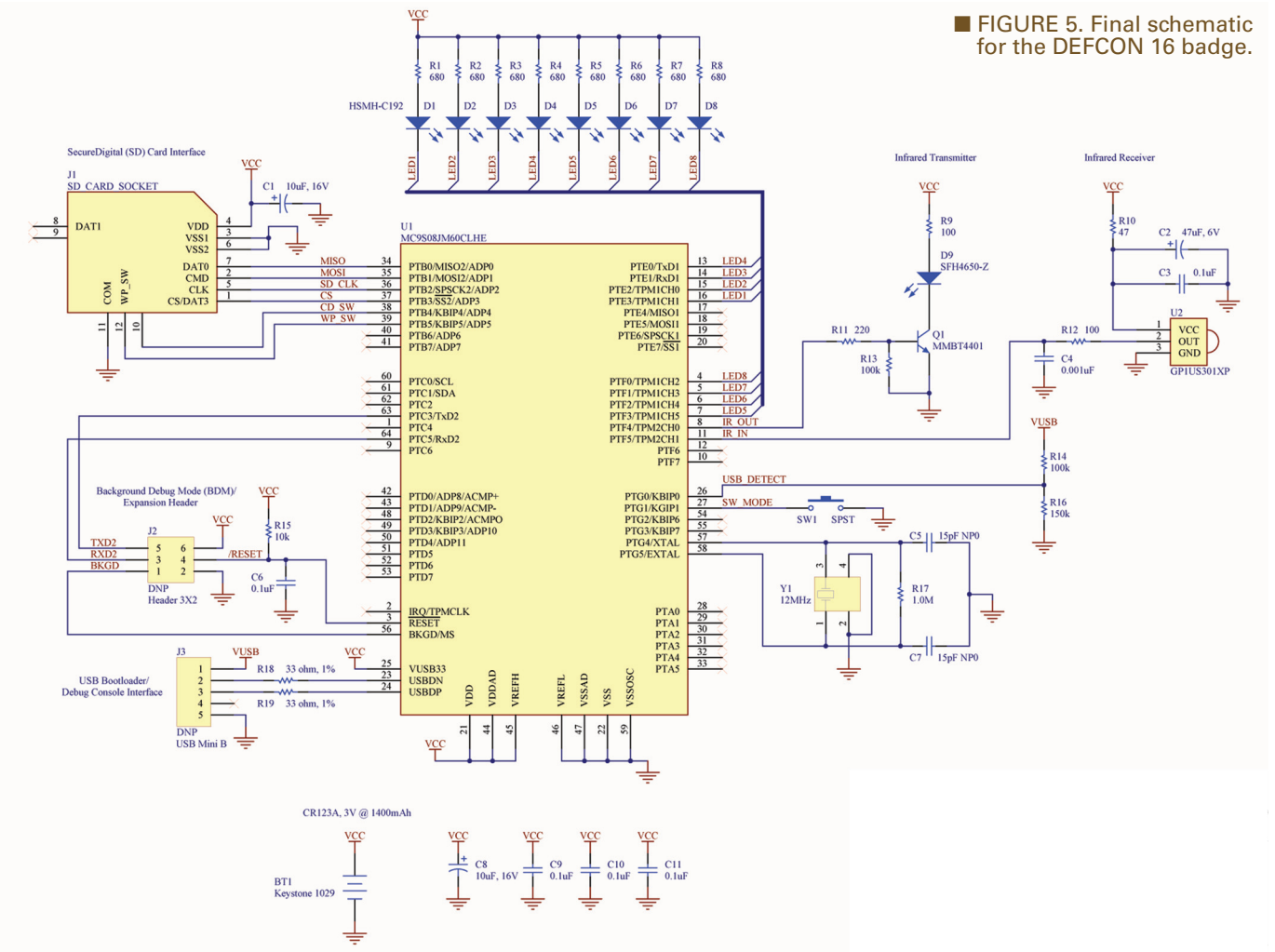
With over 8,500 attendees expected at the conference, I assumed that only a small percentage would actually take advantage of the file transfer capabilities, even though that was to be the core functionality of the badge. I wanted the badge to do something interesting right out of the box if the user didn't insert an SD card into the socket. I decided to incorporate "TV-B-Gone" functionality into the badge and take advantage of the infrared components that would already be in place. The original TV-B-Gone (**www.tv-b-gone.com**) product was designed by Mitch Altman of Cornfield Electronics. The unit simply transmits all known television remote control power-off codes one after another, allowing you to turn off practically any TV in North America, Asia, or Europe. Tens of thousands of these units have been sold over the past few years and Mitch recently released an open-source version of the product in kit form. Depending on how the TV-B-Gone is used, it can be quite mischievous and I thought it would be suitable for a hacker conference where people are used to taking advantage of and pushing the bounds of technology.

Above and beyond the engineering design goals, there were some fundamental requirements:

- A*esthetics*. The badge needed to look nice and be as non-intrusive to the wearer as possible. From the graphics to the routing to the parts placement to the circuit board traces, every aspect of the badge design was considered.
- *Low Cost*. The badges had to be cost-effective. The goal was a $7 total BOM (bill of materials) cost per unit including components, programming, PCB manufacturing, assembly, and testing for 8,500 pieces. Meeting the badge budget has been a major challenge in previous years.
- *Hackable*. The badge should be completely "hackable" in nature by providing source code, schematics, and development resources for those who wanted to modify their badge to do something different and out of the ordinary. Although any product can be hacked without provisions to do so, I wanted to make the badge welcoming to hackers and foster the hacking spirit so prevalent at DEFCON.
- *Continued Use*. The badge should be designed to provide a general-purpose development environment or reference platform that attendees can build on and learn from after the conference.

Engineering Process

With the design goals in mind, I first put together a system-level block diagram — basically a high-level conceptual drawing to help me visualize the overall design (Figure 1). This was hand-drawn and would eventually be converted into a detailed schematic later on in the process. The design is based on a Freescale Flexis



■ FIGURE 5. Final schematic for the DEFCON 16 badge.

MC9S08JM60 eight bit microcontroller (**www.freescale.com/webapp/sps/site/prod_summary.jsp?code=S08JM**) and has interfaces to the SD card socket, infrared transmission and receiver circuitry, USB port, and debug/programming connector. The JM60 microcontroller has 60KB of Flash, 4KB of RAM, a 12 channel, 12 bit ADC, USB 2.0 full-speed device support, two SPI (Serial Peripheral Interface) modules, two SCI (Serial Communications Interface)/UARTs, two timer/PWM modules, eight keyboard interrupts, real-time clock, internal reference clock, and 51 general-purpose I/Os. It's a powerful part and has lots of on-chip functionality that I could take advantage of. A set of eight LEDs on the front of the badge is used as status and mode indicators. I had over 68,000 LEDs leftover from last year's DEFCON 15 badge and wanted to do something with them.

The next step was to start developing with actual hardware. I used Freescale's off-the-shelf DEMOJM evaluation board (Figure 2; **www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DEMOJM**) and CodeWarrior Development Studio for Microcontrollers — which is freely available for up to 32KB of code (**www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CW-MICROCONTROLLERS**) — to get the basic firmware and state machine environment set up. Then, I added an SD card socket to the DEMOJM's expansion header and continued with the firmware development until I was comfortable that the intended functionality of the badge would succeed.

After that, using the system-level block diagram as a rough guide, I built a custom circuit board (Figure 3) with only the specific hardware that I wanted to have on the badge. I also designed in provisions for a few elements that I hadn't yet completely decided on, like how to support the infrared transmission and reception (discrete components or an IrDA-compliant module) and battery selection (AAA or something else). The hardware and firmware designs were finalized on this board before moving to the next step, which was a true-to-form pre-production prototype. Using an intermediary board like this allowed me to not only verify my schematic, but also to easily make changes to component values and take measurements of various signals to aid in troubleshooting and diagnostics. The majority of hardware and firmware

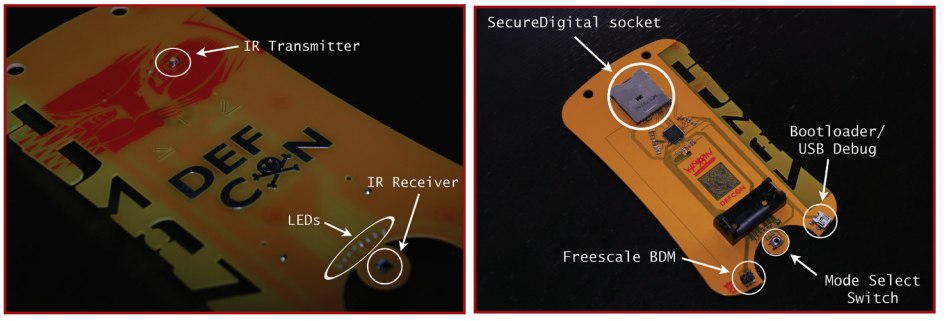
Note: Do Not Populate C4, J2, J3

■ TABLE 1. Final bill of materials (BOM).

Item	Quantity	Reference	Manufacturer	Manuf. Part #	Distributor	Distrib. Part #	Description
1	1	BT1	Keystone	1029	Digi-Key	1029K-ND	Battery holder for CR/123A
2	2	C1,C8	Kemet	T491A106M016AT	Mouser	80-T491A106M016	10uF tantalum capacitor, 16V, 20%, size A
3	1	C2	Kemet	T491A476M006AT	Mouser	80-T491A476M06	47uF tantalum capacitor, 6.3V, 20%, size A
4	5	C3,C6,C9,C10,C11	Kemet	C0603C104K4RACTU	Digi-Key	399-1096-2-ND	0.1uF bypass capacitor, 16V, X7R, 0603
5	2	C5,C7	Kemet	C0603C150J5GACTU	Digi-Key	399-1051-2-ND	15pF ceramic capacitor, NP0, 50V, 0603
6	8	D1-D8	Avago	HSMH-C192	FAI	HSMH-C192	LED, Red, 0603, 1.8Vf, 17mcd @ 20mA (leftover from DC15)
7	1	D9	Osram	SFH4650-Z	Digi-Key	475-2569-2-ND	LED, Infrared, 850nm, +/-20 degree angle, 16mW @ 100mA, SMD
8	1	J1	3M	SD-RSMT-2-MQ-WF	Digi-Key	3M5646TR-ND	SecureDigital memory socket/connector, push-push, R/A, SMD
9	1	Q1	Fairchild	MMBT4401	Digi-Key	MMBT4401FSTR-ND	Transistor, general purpose, NPN, 40V, 600mA, SOT23-3
10	8	R1-R8	Rohm	MCR03EZPJ681	Digi-Key	RHM680GTR-ND	680 ohm, 5%, 1/10W, 0603
11	2	R9,R12	Panasonic	ERJ-3GEYJ101V	Digi-Key	P100GTR-ND	100 ohm, 5%, 1/10W, 0603
12	1	R10	Panasonic	ERJ-3GEYJ470V	Digi-Key	P47GTR-ND	47 ohm, 5%, 1/10W, 0603
13	1	R11	Rohm	MCR03EZPJ221	Digi-Key	RHM220GTR-ND	220 ohm, 5%, 1/10W, 0603
14	2	R13,R14	Rohm	MCR03EZPJ104	Digi-Key	RHM100KGTR-ND	100k, 5%, 1/10W, 0603
15	1	R15	Panasonic	ERJ-3GEYJ103V	Digi-Key	P10KGTR-ND	10k, 5%, 1/10W, 0603
16	1	R16	Panasonic	ERJ-3GEYJ154V	Digi-Key	P150KGTR-ND	150k, 5%, 1/10W, 0603
17	1	R17	Panasonic	ERJ-3GEYJ105V	Digi-Key	P1.0MGTR-ND	1.0M, 5%, 1/10W, 0603
18	2	R18,R19	Yageo	RC0603FR-0733RL	Digi-Key	311-33.0HRTR-ND	33 ohm, 1%, 1/10W, 0603
19	1	SW1	C&K	KSC341JLFS	Digi-Key	401-1770-2-ND	SPST tactile momentary switch, 300gf, 6.2mm x 6.2mm, SMD
20	1	U1	Freescale	MC9S08JM60CLH	FAI	MC9S08JM60CLH	Microcontroller, LQFP64
21	1	U2	Sharp	GP1US301XP	Digi-Key	425-2527-2-ND	Receiver Module, Infrared (IR), 38kHz, 2.4V-5.5V, SMD
22	1	Y1	NDK	NX3225SA-12.000000MHZ	Digi-Key	644-1047-2-ND	Crystal, 12MHz, 8pF, SMD
23	1	PCB	e-Teknet	DC16 1.0	N/A	N/A	PCB (includes assembly and testing)

development was done on this platform before moving to the "form and function" pre-production prototype.

With the hardware and firmware completed, the final task was to lay out the actual badge circuit board and build a few pre-production prototypes to verify the entire system before kicking off the production run (Figure 4). I ordered a few bare boards with yellow soldermask and red silkscreen (a color combination I had never seen before and was curious as to what it would look like), hand-soldered them, and ran through my test procedure to verify that the individual aspects of the badge worked as desired. This step was the last chance for me to correct any mistakes before committing to many thousands of dollars of circuit boards and components. I also used these prototypes as samples for DEFCON to approve.

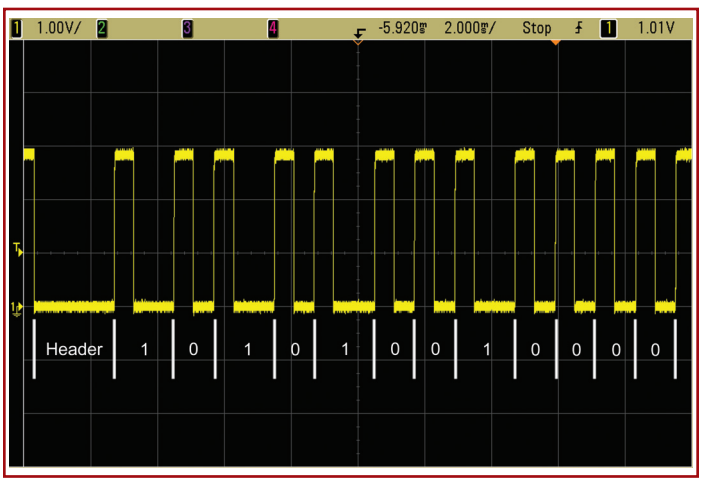


■ FIGURE 6. Final PCB design showing major subsystems.

Using two hand-soldered pre-production prototypes let me verify that the complete design looked and worked as desired. Each board took about one hour to assemble.

Figure 5 and Table 1 show the final schematic and bill of materials, respectively. The total BOM cost per unit was $10.72, not including taxes or shipping. The largest line items were the PCB (printed circuit board) fabrication, manufacturing, assembly, and testing at $3.88 and the microprocessor at $1.95 (Freescale gave us a large discount on these parts, as they are normally priced at $3.05 for 1,000 piece quantities).

What I thought would be the hardest part of the project (the engineering) was completed with relatively few mishaps.



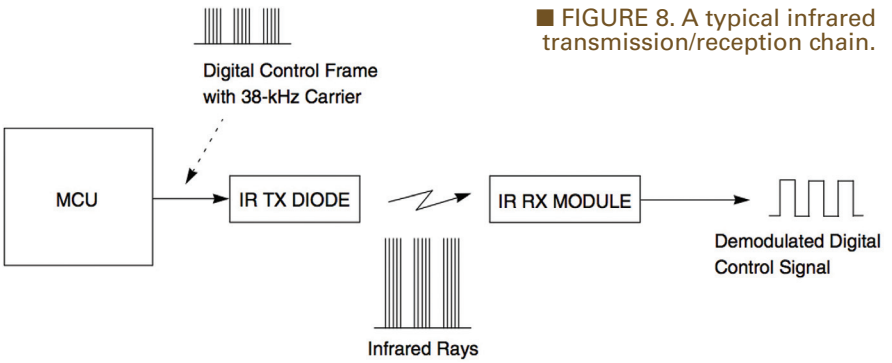
■ FIGURE 7. A Sony TV power off code. I captured this signal from an IR remote control receiver module, so it is inverted.

Badge Functionality

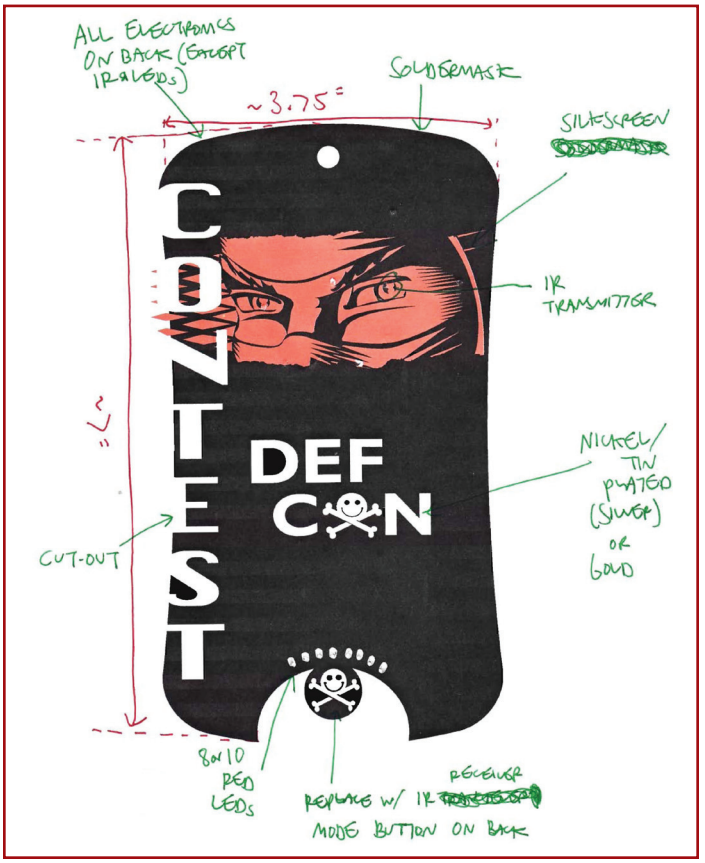
The DEFCON 16 badge packed in lots of functionality using minimal components (Figure 6). A single pushbutton switch serves as the user interface to cycle through the badge's three operating states:

- Receive file
- Transmit file (or TV-B-Gone if no SD card is inserted
- Sleep

Let's take a look at the technical details of the major



■ FIGURE 8. A typical infrared transmission/reception chain.



■ FIGURE 9.Concept sketch of the PCB with my hand-written notes.

■ FIGURE 10. The eight different DEFCON 16 badge styles.

subsystems of the badge.

Infrared Remote Control and TV-B-Gone

For the infrared (IR) subsystem, I decided on using on-off keying — one of the oldest and simplest modulation techniques. I would essentially turn on and off a low frequency carrier (in our case, 38 kHz) in order to modulate data. Then, using an encoding scheme known as Pulse-Width Encoding, I defined a logic 0 and logic 1 by the width of the "on" pulse, while the pulse distance (the distance in between the pulses) or "off" pulse remained constant. This no-frills approach is used by just about every infrared remote control device on the market and the only circuitry I needed was an infrared LED (D9), current-limiting resistor (R9), and transistor driver (Q1, R11, R13).

I opted for this discrete approach over using an IrDA (**http://en.wikipedia.org/wiki/Infrared_Data_Association**) transceiver module — a common, robust solution for file transfer between computers or consumer devices — for a number of reasons. Many IrDA modules (such as the Vishay TFDU4300) require an additional encoder/decoder (whether done in hardware, like the Vishay TOIM4232, or software) to convert serial data to IrDA-compatible pulses. I also felt that a discrete solution would be more hackable, as one could modify the firmware to generate any sort of IR transmission desired, instead of being forced to adhere to the imposed standards of the IrDA module. Cost was a concern, as well, and I was able to implement the discrete IR circuit for $1.48 versus approximately $6 I would have had to pay for a fully IrDA-compliant design.

I used one of the JM60's timer/PWM channels to generate a 38 kHz carrier at a 33% duty cycle and could turn the carrier on or off by simply enabling or disabling the PWM channel. As an initial infrared test, I decided to impersonate a Sony TV power off code to see if I could turn off my television using my badge development board. The Sony remote control specification is well documented online and defines a logic 1 as a 0.6 ms off pulse followed by a 1.2 ms on pulse, and a logic 0 as a 0.6 ms off pulse followed a 0.6 ms on pulse. I simply duplicated the entire pulse train for a power-off signal (Figure 7), not caring about what data I was actually transmitting. The test worked perfectly! Now, I could move on to incorporating the TV-B-Gone functionality. I captured this signal from an IR remote control receiver module, so it is inverted.

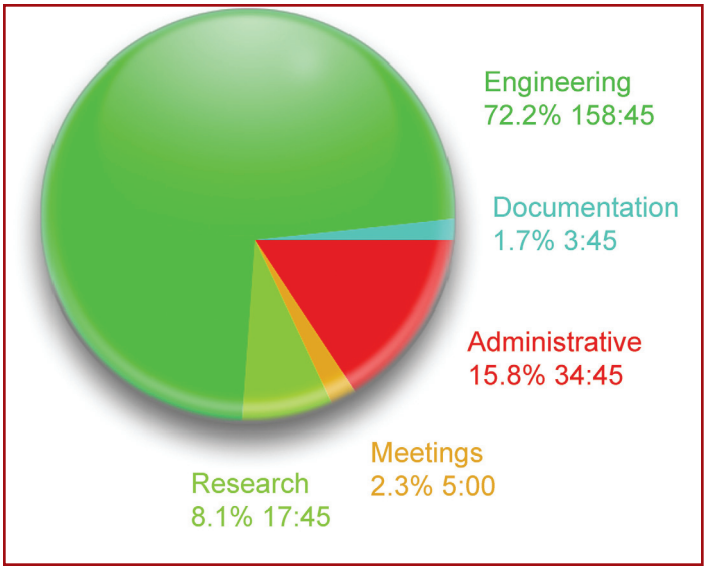
The TV-B-Gone simply transmits all known television remote control power-off codes at their pre-defined carrier frequency and pulse-width timings, one after another. The open-source version of the TV-B-Gone (**www.ladyada.net/make/tvbgone**) contains a header file with all of that information. I grabbed the header file and ported the

TV-B-Gone functionality to the badge by parsing the data, properly configuring the PWM channel, and turning the carrier on and off at the correct timing. The IR LED (Osram SFH4650-Z, D9) that I selected for the badge is low power and narrow beamwidth (± 20 degrees half angle). The narrow beamwidth is especially important for the file transfer mode to prevent interference between multiple people transferring files within the same area. Because of that, the TV-B-Gone functionality only works with televisions a few feet away. Most attendees who really wanted to take full advantage of the TV-B-Gone mode replaced the stock IR LED with a high brightness, wide beamwidth LED to get the farthest range possible.

Infrared File Transfer with SecureDigital Card and FAT File System Support

As opposed to the transmit-only functionality of the TV-B-Gone mode, a file transfer requires one badge to transmit and one badge to receive. The badge uses a Sharp GP1US301XP infrared receiver module for remote controls, which is tuned to 38 kHz (the same as my IR transmit modulation frequency). The receiver will bandpass the incoming signal to help reduce noise caused by the ambient environment (in particular, lighting) and then provide a demodulated signal at logic levels that can easily be interfaced with a microprocessor. Figure 8 shows a typical IR transmit and receive chain.

The goal of the file transfer feature is to read a file from an SD card, transmit it to a willing recipient, and store the received file onto the recipient's SD card. The physical interface from the SD card socket to the JM60 microprocessor is as simple as it gets. In its most basic configuration, SecureDigital uses an SPI interface, consisting of four lines — Master In Slave Out (MISO), Master Out Slave in (MOSI), Clock (CLK), and Chip Select (CS) — for its MultiMediaCard (MMC) protocol. Two additional switches on the socket — Card Detect (CD) and Write Protect (WP) — are connected to two general-purpose inputs on the processor. It is trivial to read and write data to the SD card using SPI, as it's essentially just an external serial memory device, but the trick is incorporating the FAT file system structure (**http://en.wikipedia.**



■ FIGURE 11. Time breakdown of the DEFCON 16 badge project. Thankfully, most of my time was spent doing what I love to do — engineering.

DEFCON 16 Badge Care and Feeding Guide

Hackers thrive on figuring things out on their own, so I wrote a somewhat cryptic "guide" for the DEFCON 16 conference program to get people started using the badge without giving away too much information.

- Insert battery.
- Badge starts up in Receive mode.
- Press button to change modes.
- Next mode is Transmit mode.
- If no SD card inserted, enters TV-B-Gone mode.
- Turn off all TVs in range.
- Hack IR LED for wider propagation and higher brightness.
- Next mode is Sleep mode.
- Zzzzzzz.
- Wake up with button press.
- Insert SD cards into two badges.
- SD card must be formatted as FAT16.
- Desired file to transmit must have read-only bit set and in / directory.
- Maximum file transfer size intentionally limited to 128KB.
- Hold one badge up to another badge.
- Enter Transmit mode on one badge.
- Transfer data via IR at a speedy 771 bits per second.
- When progress bar finishes filling or emptying, transfer done.
- The further away you are, the less likely it will work.
- Transfer will abort if bad CRC, no data received, or button pressed.
- Trade wares with other hackers.
- Enter Badge Hacking Contest.
- Look at source code, schematics, and other badge inf0z on DEFCON CD.
- Modify firmware.
- Modify hardware.
- Modify badge.
- Impress Kingpin to win prizes.
- Spend time in the Hardware Hacking Village.

Own hotel TVs or control your BSODomizer with infrared.

- Battery will last way longer than DEFCON does.
- LEDs make nice patterns.
- Go to **www.kingpinempire.com**.

org/wiki/File_Allocation_Table) so you can load and retrieve files from any computer system.

There are lots of available implementations of FAT for embedded systems and it didn't make sense for me to try and recreate the file system from scratch. As luck would have it, Freescale was in the midst of creating a small reference design — a data logger with light sensor, USB, SD card, and FAT file system. It wasn't publicly released, but they were gracious enough to share their source code with me, which gave me a huge head start on getting my implementation working. Freescale's design only supports SD cards that have been formatted in FAT16, which means it will only work with cards 64 MB or larger. Windows automatically formats cards less than 64 MB as FAT12, which will lead to a corrupted FAT table on the SD card if used in the badge. I learned this the hard way after countless hours of troubleshooting.

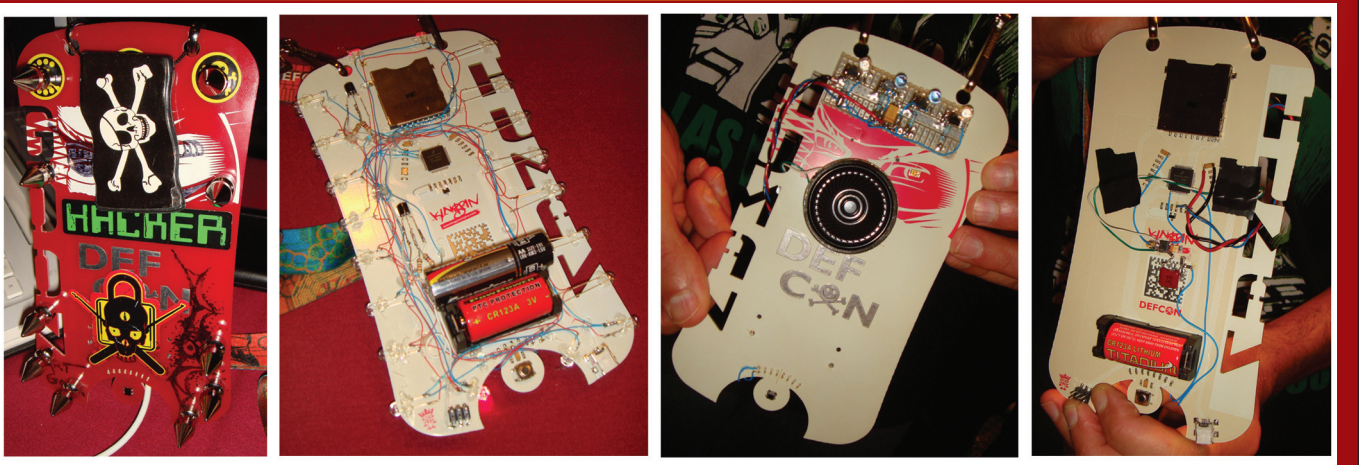
With the low-level SD card and FAT file system support complete, I could move on to designing the actual file transfer mechanism. Instead of making use of an existing file transfer protocol like Kermit or XMODEM, I decided to roll my own version, which would be more educational (though much less graceful). I decided to build on the same on-off keying and pulse-width encoding schemes that were used for my initial infrared transfer tests with the Sony protocol, and added a mechanism to send large streams of data instead of short remote control codes. I reduced the pulse width timing in order to increase data transmission speed and added a way to send the filename, file size, and CRC along with the actual file. The file transfer happens as follows:

- Load the file you want to transmit onto the host's SD card. The file must only have the read-only flag set — that way, the badge will know that's the file you want to transmit and not a file you've received from someone else.
- Set the target badge to Receive mode (the LEDs on the front of the badge will move left to right and right to left like KITT from Knight Rider).
- Set the host badge to Transmit mode (the LEDs will illuminate from the center outwards).
- Hold the badges a few inches apart from each other and the file transfer will begin. Data is transmitted serially at 771 bits per second (one byte every 10.375 ms). The

DEFCON 16 Badge Hacking Contest

The DEFCON 16 badge is essentially a wearable reference platform and the possibilities are limitless to what can be done with it. For the third year in a row, I hosted the Badge Hacking Contest to see what sorts of ingenious, obscure, mischievous, or technologically astounding badge modifications can occur during the weekend-long DEFCON. The impetus is simple: Get people excited about electronics, learn something new, share your work with others, win prizes.

A key part of the Badge Hacking Contest is the environment in which it takes place. As opposed to DEFCON 15, where I set up a simple folding table in the DEFCON vendor area with a single soldering iron, some extra components, and development tools, this year's Badge Hacking "Headquarters" was set up in the new Hardware Hacking Village (HHV) — a separate room reserved for those interested in exploring and experimenting with electronics, filled with soldering irons, development tools, and other equipment, and hosted and supported by volunteers eager to share their passion with others. Just like last year, Freescale was gracious enough to send out a real-life engineer to hang around and support the badge hacking endeavors by answering questions and providing technical details of the Freescale JM60 microcontroller that was used on the badge. The HHV allowed people with like-minded interests to gather in a more intimate environment and the response was greater than any of us could have imagined. The room was completely packed throughout the weekend, with hundreds of curious attendees stopping by to attempt to solder the



■ FIGURE 12. A montage of Badge Hacking Contest entries.

host badge first transmits the filename and file size. If the filename already exists on the target SD card, the name will increment automatically to prevent duplicates. I also set an intentional 128KB file size limit, since it's impractical to transfer that much data using this method and it gives me an upper bound for error checking.

• Finally, the actual file will be sent. The LEDs serve as status indicators on both badges, with the LEDs on the host badge starting off all illuminated and turning off one by one as the file is sent, and the target badge starting with all LEDs off and turning on one by one as the file is received. A CRC-16 checksum is sent after every 512 byte block. Because 512 bytes is the standard block size for an SD card, it made sense to just send the entire block followed by the checksum instead of trying to split up the block. If the CRC matches what the receiver has calculated for that given block, then the transfer continues. Otherwise, the entire transfer is aborted (the LEDs alternate in a pattern alerting the user of the failure and then the badge goes to sleep).

It's not the most graceful approach, but it worked well enough!

USB Debug Console and Bootloader

In previous years' designs, the only way a badge user could load new firmware onto the badge was through whatever proprietary programming/debugging interface was provided by the selected microprocessor. DEFCON 14 used the Microchip PIC ICD2 interface and DEFCON 15 used Freescale's BDM (Background Debug Mode) interface, and both required specialized hardware. I think that hindered participation in previous Badge Hacking Contests, as I only brought a few programming units that people would have to share.

The USB interface on the DEFCON 16 badge serves two distinct purposes. Most important is the bootloader functionality, described in Freescale's application note AN3561 (**www.freescale.com/files/microcontrollers/doc/app_note/AN3561.pdf**), that allows in-circuit reprogramming of the JM60's Flash memory. With the bootloader enabled (achieved by holding down the mode select button while

mini-USB connector onto their badge or just to perform other hardware- and badge-hacking activities.

There were 20 official entries in this year's contest, up from seven the year before, ranging from pure hardware or firmware modifications to complicated combinations of both (Figure 12). The top three winners are detailed here, along with a sampling of other contest entries. Due to the success of the Badge Hacking Contest, it has become an official "Black Badge" event, meaning that the winner receives a coveted Black Badge good for lifetime entry to DEFCON. So, warm up your soldering irons and start planning for next year's contest. Hope to see you there!

1ST PLACE: Human Password Generator by the Greek Geeks.

A software application on a PC laptop tracked the motion of the badge's LEDs via a webcam and sent a hash of the motion profile over USB to the badge, which then computed the password based on the motion hash and transferred the result back to the PC.

2ND PLACE: Apple Front Row and HP Pavilion DV Laptop remote control emulation by BonzoESC, Sterling, Critta, and Jymbolia.

This hack uses the badge's IR transmitter to emulate Apple Front Row and HP Pavilion DV-series laptop remote controls. Also provides brute-forcing of the eight-bit keyspace used for pairing of a Front Row remote control to computer. More details available from **http://github.com/bkerley/dc16_badge**.

3RD PLACE: Motion-generated music and snooze alert (tilt detection) using accelerometer.

Motion-generated music using an external accelerometer connected to the JM60. Also features "snooze alert" by making sound if a tilt is detected.

OTHER SELECTED ENTRIES (in no particular order):

* Cellular automaton with two models: Rule 30, random number generator; and Rule 110, Turing-complete "spaceship pattern," displayed on the LEDs.
* Optical trojan/covert channel via LED (transmitting "HELLO DEFCON" in Morse code, undetectable to the human eye.)
* Real-time binary clock with time synchronization between two badges via IR.
* Persistence-of-vision (POV) saying "DC16 HACKER" and Pseudorandom number generator w/ PWM dimming.
* Nikon camera remote control/trigger via IR.

applying power to the badge), the user simply needs to connect their badge to a PC with the mini-USB connector and load their compiled code onto the badge using a free GUI. The only downside is that no debugging capability exists through the bootloader. If, for some reason, the firmware gets corrupted through a faulty programming operation, a standard six-pin BDM header is also made available on the badge for complete re-programming.

The USB bootloader satisfied our requirement of a hackable badge. Since no specialized development hardware was necessary to modify the firmware of the badge, the pool of potential badge hackers increased tremendously.

When not being used by the bootloader, the USB port serves as a virtual serial port created using the standard USB Communications Device Class (CDC). With the proper driver installed (which was included on the DEFCON CD), you can simply load up your favorite terminal program, and transmit and receive serial data. I used this feature during development to send debug messages and left the capability enabled for attendees to explore and use for their own hacks.

Power/Batteries

The selection of power supply components and batteries was a key challenge of this design. I did not want to repeat the problems of last year's badge, in which the batteries would be depleted within two days if the badge was heavily used. There were five elements I needed to be concerned with, in order of priority:

1. Battery Life (must last longer than the weekend-long DEFCON)
2. Availability
3. Cost
4. Complexity
5. Weight

I began by taking current measurements of the major operational states of the badge on my custom development board (all @ 3V):

- Sleep = 0.79 mA
- IR Receive mode = 5.3 mA
- IR Transmit mode (continuous data transmission) = 9.1 mA
- SD Card (continuous read and write) = 25-35 mA, but can be as high as 200 mA, according to SecureDigital specification

When the USB connection is plugged in, the badge automatically increases the microcontroller's clock speed from 12 MHz to 48 MHz (required for the JM60's USB module to function), so current consumption increases across the board by about 20 mA.

My previous DEFCON badges used one or two 3V CR2032 Lithium coin cells, which are lightweight and low profile. However, they aren't suitable for high-current applications (greater than 3 mA continuous or 10 mA pulse), so they couldn't be used for this year's design. The original plan was to try and power the badge from a single AAA battery. After looking into some single-cell boost converters/regulators— such as the Micrel MIC2571, Sipex SP6644, and Sipex SP6641B — to boost the 1.5V nominal battery cell voltage to a higher system voltage, I decided that, although suitable for many consumer electronic devices, the additional external components (such as inductors and specialized capacitors), critical PCB layout, and potential switching noise introduced into the system, would have made this direction too risky to use on the badge with such a short development cycle.

Next, I had narrowed down the selection to either three AAA batteries or a single CR123A Lithium cell. Both were easily available at convenience, photography, and electronics stores, though the AAAs are arguably more common and definitely cheaper. It was now apparent that whatever battery solution we went with would be heavier and more bulky than we would have liked, but my primary concern was making sure the system would function as designed, and I'd just brace for any negative comments (of which there were very few). I spent hours on the phone with the DEFCON organizers discussing battery chemistry and the pros and cons of each battery type. They chose to move forward with three AAA batteries, which would provide 4.5V at 1,250 mAh that I could bring down to a 3V system voltage with a low-cost linear regulator. But, when I started to look around to purchase 25,500 surface-mount AAA battery holders, there was not enough stock worldwide and the manufacturer's leadtime was past the date of DEFCON, which killed this approach on the spot.

We settled on using the CR123A, which I now know is a better, simpler solution. The battery has built-in PTC (Positive Temperature Coefficient) protection to limit current flow in a short circuit or battery failure condition, and doesn't require any external voltage regulation circuitry in order to be used in our system (thanks to the stiff voltage output the battery has until it's close to end-of-life). The above current measurements show that with a single 3V CR123A rated at 1,400 mAh, the badge can last for hundreds of hours of normal use, satisfying our requirement that the badge remains operational for the length of DEFCON.

Circuit Board

Creating the badge electronics was only one part of the battle, as aesthetics of the badge was also a fundamental design goal. Each year when the folks at DEFCON say "Let's try to make a badge that looks like this," they're saying it purely from an artistic point of view. They're not concerned with any electrical characteristics, manufacturing methods, or PCB related limitations or constraints. That naivety is what pushes me to try PCB design and layout techniques that I wouldn't normally consider in order to meet their proposal. Figure 9 shows the initial badge concept sketch sent to me by the DEFCON organizers. I scribbled some of my ideas onto

the page and sent it back to them for approval. I also ran my ideas by e-Teknet (who handled the board fabrication and assembly) to make sure we could actually fabricate the badges in production quantities and do so in a cost-effective manner.

There were a number of major elements to the badge's circuit board design: the physical board outline's subtle curves; the complicated text cut-outs for the conference attendee type; masking certain areas of soldermask to bring out graphics on the copper layer; and parts placement, such as the arc of LEDs along the bottom edge and locating the IR transmitter in the ninja's left eye. I made a conscious decision to leave all parts designators off of the badge. This lends itself to a much cleaner look at the expense of easy parts identification. The assembly drawings were available to people curious about hacking or modifying the badge to make their job easier.

I had added in a few surprise graphical elements, such as a two-dimensional Data Matrix barcode (**http://en.wikipedia.org/wiki/Datamatrix**) and a secret message used for the Mystery Challenge — a popular hardware hacking/puzzle contest at DEFCON.

A total of 8,500 badges were manufactured and eight different text cut-outs and soldermask/silkscreen color combinations were used to denote the different DEFCON clientele (Figure 10):

Human: White soldermask/Red silkscreen, 7,500 pieces.

Goon (DEFCON aide): Red soldermask/White silkscreen, 200 pieces.

Staff: Red soldermask/White silkscreen, 150 pieces.

Press: Green soldermask/White silkscreen, 150 pieces.

Speaker: Blue soldermask/White silkscreen, 250 pieces.

Vendor: Purple soldermask/White silkscreen, 100 pieces.

Contest Organizer: Yellow soldermask/Red silkscreen, 50 pieces.

Uber (Awarded to the winners of official DEFCON contests): Black soldermask/Yellow silkscreen, 100 pieces.

Supply Chain and Manufacturing Problems

Sourcing and obtaining components for a production build is never easy, as there are many potential pitfalls along the supply chain. Things like lack of (or misquoted numbers of) available stock, long leadtimes, shipping delays and mishaps, and other human errors can cause electronics production to grind to a screeching halt. The technical design portion of this project went relatively smoothly, but the supply chain and manufacturing problems we encountered led us to create 10,000 temporary plastic badges and threatened to cause the entire electronic badge to be cancelled for DEFCON 16.

Parts sourcing was the first major complication. Knowing about the finality of our deadline, I began ordering parts as soon as I could. Even still, trying to find 8,500 pieces of anything is hard. If I couldn't get parts in hand quickly, I'd redesign with a part that was available. Most of the components — such as discretes, batteries, switches, and connectors — were received early enough.

The SecureDigital socket is one part that didn't arrive on schedule and with the crux of the badge being the file transfer and SD card support, the part was an absolute necessity. After evaluating a number of SD card sockets, I selected the 3M SD-RSMT-2-MQ-WF, primarily because it was half the price of any competing socket, Digi-Key had a bit of stock (1,200 pieces), and the leadtime for the balance was quoted at 2-4 weeks less than the other manufacturers. I placed the quantity order in May through Digi-Key and 3M had promised to deliver the remaining 7,500 pieces in six weeks, which gave us plenty of leeway before we were to start production manufacturing. Immediately after placing the order, the leadtime was increased to 8-10 weeks. That would be cutting it very close to our production deadline and should have been a major red flag, but I naively assumed 3M would live up to their delivery promise and took the risk to wait.

Eight to 10 weeks pass. The boards have been manufactured. The microprocessors have been programmed. The entire design is locked in. Then, on July 16, I get the call. "The sockets aren't done." My heart jumped into my throat, and when I asked why, the only response I got was, "I don't know." After a few days and numerous phone calls, 3M was able to commit to a delivery date of August 8th. What? That's the first day of DEFCON! Obviously an unacceptable answer, I was fuming and maybe it was my relentless poking and prodding that I finally received a call from 3M's Global Account Manager. If anyone could solve this problem, it was him. He said, "I've got a handle on it, I'll get the parts to you right away." And, as if nothing had gone wrong, the parts arrived to e-Teknet in China 10 days before DEFCON. Apparently, there was a miscommunication and the stock had been sitting somewhere in Singapore. To this day, I still don't know the real reasons behind the mishap and at the time I didn't care. I had the parts and we were finally ready for manufacturing. Or, so I thought.

Immediately after the SD card socket debacle had come to an end, I received a phone call from e-Teknet informing me that one last box of components I had sent to them had been stuck in Chinese Customs for five weeks. That box just happened to contain critical parts (IR receivers, IR LEDs, and crystal oscillators, all with unique surface-mount footprints) that e-Teknet couldn't obtain locally. It turns out that the first day of DEFCON 16 coincided with the first day of the 2008 Olympics in Beijing and everything coming into and out of China for months prior was being examined and heavily scrutinized. e-Teknet was hoping that the box would get released in time to start manufacturing, but it wasn't to be, and we had to scramble to try to get these parts through. As with the SD card socket, the IR components were paramount to the file transfer capabilities of the badge and without them, no files would be transferred. To make matters worse, without the IR LEDs, even the standard TV-B-Gone

feature would not be operational.

e-Teknet tried everything they could from within China to get the parts released from Customs, all to no avail. Once parts are held, it can be months before they're released. I placed two more orders with Digi-Key (luckily, they had more than enough stock): one to ship directly to China and one to ship to e-Teknet's local office in Arizona. The box sent to China was also held in Customs. Something about these components was obviously drawing attention, but I had no idea what. In one last ditch effort, e-Teknet in the US cut up the order into a bunch of smaller pieces and sent them in multiple boxes to try and get through Customs. I spent every waking moment clicking on the UPS tracking numbers to follow their progress. On August 3rd — four days before DEFCON registration opened — all of those smaller boxes magically passed through Customs and e-Teknet had everything they needed to begin. Yes, you read that correctly. All 8,500 badges now had to be assembled, tested, and shipped to Las Vegas in four days! e-Teknet would be using an automated process of high-speed pick-and-place machines and reflow ovens with minimal human intervention (except for those people running the machines and handling final test and inspection). As for the two boxes that were still stuck in Customs, we requested them to be released and returned to sender, which they eventually were.

The final little snag came during manufacturing. Instead of starting some portion of manufacturing while waiting for the balance of parts to come in, e-Teknet had made a decision to wait for all the parts before beginning. I'm sure there were valid business reasons for doing so, since it would have taken significant resources and machine time to only partially manufacture badges and then have to go back later to complete them (either automatically or by hand). They were confident that they could still deliver the badges in time which would be a miraculous feat, barring any unforeseen glitches.

Unfortunately, there was a glitch. e-Teknet was having trouble with the optical alignment of the badge PCBs in their pick-and-place machine, which required three fiducial markers for accuracy. This shouldn't have been a problem, since we had enough bare copper pads near the corners of the boards that could have been used as fiducials, even though we hadn't specifically designed any in (I will next time!) The lack of automatic alignment drastically slowed down progress. At this point, not even e-Teknet thought the boards would be done in time. They set up another machine to have two going in parallel around the clock and I was in touch with the factory multiple times a day. I was already in Las Vegas at this point preparing for DEFCON and got the necessary people involved to update them on progress. As a backup plan, DEFCON's resident artist began a design of a temporary plastic badge that would be distributed to all the attendees who had not yet received or were unable to receive an actual electronic badge. We had to cover our bases as best we could in case the badges didn't show up.

Each day, e-Teknet sent a few thousand pieces as they came off the assembly line. When all was said and done, we received all 8,500 badges, with the final box arriving on Sunday (the last day of the conference), much to the joy of a few thousand more attendees that got to receive a real badge. International expedited shipping costs ran upwards of $14,000, which was obviously an unforeseen hit to our overall badge budget. With these charges, the cost savings of outsourcing manufacturing to China had been negated and we could have avoided the Customs problems by simply remaining in the US and paying a higher manufacturing rate.

Most people never think about where a conference badge comes from or how it was designed. This year was different. The delay in badges was a topic of conversation throughout the weekend. As the badges came trickling in, long lines formed in front of the DEFCON registration area for people to swap out their temporary plastic badge for the real deal. Attendees missed talks and opportunities to hang out with friends in order to stand in line for hours. The pressure was real to try and get these badges into every attendee's hands. While I'm happy and relieved that it all worked out, I was somewhat embarrassed at the situation and have chalked this up as one big learning experience.

Until Next Year ...

All told, the DEFCON 16 badge project took about 220 hours, including the firefighting of supply chain and manufacturing problems after development was completed. The majority of engineering was done on nights and weekends, much to the chagrin of my very pregnant wife, Keely, as I was then spending my days as a co-host of Prototype This (**www.discovery.com/prototypethis**). Figure 11 is an interesting visualization of the time dedicated to each aspect of the project.

Thankfully, most of my time was spent doing what I love to do — engineering.

We've already started to think about designs and features for the DEFCON 17 badge and have had a lot of great input from attendees and badge hacking contest participants. Next year's design will be even simpler and more accessible to electronics hobbyists and beginner hardware hackers, but will also contain some "so new it doesn't even exist yet" technology to impress even the most hardened gadget geek. Even though I'll be keeping the lessons of previous years' badges in the back of my mind, I can all but guarantee there will be unexpected problems this time around. Who ever said engineering was boring?!

Complete source code, schematics, audio, video, and other documentation for the DEFCON 16 badge is a vailable at **www.grandideastudio.com/portfolio/defcon-16-badge** NV